Tunneling Internet protocols inside QUIC

draft-piraux-intarea-quic-tunnel
draft-piraux-intarea-quic-tunnel-session
draft-piraux-intarea-quic-tunnel-tcp

Maxime Piraux, Olivier Bonaventure, Adi Masputra

This work was partially supported by the MQUIC project
Content

- **Introduction to QUIC**
- The tunnel mode
  draft-piraux-intarea-quic-tunnel
- The tunnel session mode
  draft-piraux-intarea-quic-tunnel-session
- The stream mode
  draft-piraux-intarea-quic-tunnel-tcp
- Conclusion
Introduction to QUIC

- Provides services equivalent to TCP+TLS atop UDP
- 1-RTT Authenticated Handshake
  - 0-RTT “session resumption”
- All application data and most control data is encrypted
  - Immune to middleboxes interference, and likely to pass through given its adoption for the web
- Two manners of conveying application data
  - Streams: reliable, in-order, uni- and bi-directional bytestreams
  - Datagrams: unreliable messages
- QUIC seems like a good fit for an alternative to IPSec, (D)TLS tunnels
Content

- Introduction to QUIC
- The tunnel mode
  draft-piraux-intarea-quic-tunnel
- The tunnel session mode
  draft-piraux-intarea-quic-tunnel-session
- The stream mode
  draft-piraux-intarea-quic-tunnel-tcp
- Conclusion
Reference environment

- Client uses a Concentrator to convey its traffic over the access network.
Reference environment

- Client uses a Concentrator to convey its traffic over the access network.
- Client sends all its packets to the Concentrator over a QUIC connection.

Legend:
--- QUIC tunnel connection
=== Tunneled flow
Reference environment

- Client uses a Concentrator to convey its traffic over the access network.
- Client sends all its packets to the Concentrator over a QUIC connection.
- The Concentrator forwards them to their final destination.
- Returning traffic destined to the Client is sent over the QUIC connection
The tunnel mode

- Negotiated using the “qt” ALPN token.
- Packets are transmitted inside QUIC DATAGRAM frames.
- Out-of-band signalling is used to negotiate the type of exchanged packets.
The tunnel mode

- **Access Report TLV (AR TLV)** allows to report access network availability status.
- The Client can signal when the network is unstable to stop incoming data.
- Later, it can resume the use of the QUIC tunnel connection in the same manner.
Content

- The tunnel mode
draft-piraux-intarea-quic-tunnel
- **The tunnel session mode**
draft-piraux-intarea-quic-tunnel-session
- The stream mode
draft-piraux-intarea-quic-tunnel-tcp
- Conclusion
Reference environment

- Client uses a Concentrator to convey its traffic over the access network.
- Client sends all its packets to the Concentrator over a QUIC connection.
- The Concentrator forwards them to their final destination.
- Client is often multihomed and/or multistack, e.g. WiFi and 5G, IPv4 and IPv6.
- Client would like to leverage both access networks.
  - e.g. for load-sharing or fail-over

Legend:
--- QUIC tunnel connection
=== TCP/UDP flow
Client uses a Concentrator to convey its traffic over the access network.

Client sends all its packets to the Concentrator over a QUIC connection.

The Concentrator forwards them to their final destination.

Client is often multihomed and/or multistack, e.g. WiFi and 5G, IPv4 and IPv6.

Client would like to leverage both access networks.
  ○ e.g. for load-sharing or fail-over
Client uses a Concentrator to convey its traffic over the access network.

Client sends all its packets to the Concentrator over a QUIC connection.

The Concentrator forwards them to their final destination.

Client is often multihomed and/or multistack, e.g. WiFi and 5G, IPv4 and IPv6.

Client would like to leverage both access networks.
  ○ e.g. for load-sharing or fail-over
The tunnel session mode

- Connections can be grouped into a QUIC tunnel session.
- Allows coordinating packet reordering across connections.
- An optional opaque value indicates the QoS requested for each connection.
The tunnel session mode

- Explicitly identifies L2 or L3 packets exchanged over the QUIC connection.
- Packets are encoded inside a QUIC DATAGRAM frame using the format:

```
+---------------------------------+-
<table>
<thead>
<tr>
<th>Protocol Type (16)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet (*)...</td>
</tr>
</tbody>
</table>
+---------------------------------+-
```

- Protocol Type: The protocol type, as in the "ETHER TYPES" IANA registry.
- Packet Tag: An opaque value. It can be used for reordering.
The tunnel session mode

- Connections can be grouped into a QUIC tunnel session.
- Allows coordinating packet reordering across connections.

Legend:
- --- QUIC tunnel connection A
- --- QUIC tunnel connection B
Content

- The tunnel mode
- The tunnel session mode
  draft-piraux-intarea-quic-tunnel-session
- **The stream mode**
  draft-piraux-intarea-quic-tunnel-tcp
- Conclusion
Encapsulation overhead

- Many protocols can be conveyed using these approaches.
- But it implies a significant byte overhead.
- We introduce another operating mode dedicated for conveying TCP bytestreams.
Encapsulation overhead

- Many protocols can be conveyed using these approaches.
- But it implies a significant byte overhead.
- We introduce another operating mode dedicated for conveying TCP bytestreams.

```
P. Type  <. tunnel
P. Tag    </ session mode
----------------- <-.
    |  IP  |  |
----------------- Tunneled
    |  TCP |  |
----------------- TCP packet
    ....  <-.
```
Encapsulation overhead

- Many protocols can be conveyed using these approaches.
- But it implies a significant byte overhead.
- We introduce another operating mode dedicated for conveying TCP bytestreams.
Encapsulation overhead

- Many protocols can be conveyed using these approaches.
- But it implies a significant byte overhead.
- We introduce another operating mode dedicated for conveying TCP bytestreams.
## The stream mode

<table>
<thead>
<tr>
<th>Client</th>
<th>Concentrator</th>
<th>Final Destination</th>
</tr>
</thead>
</table>

Legend:
- --- QUIC connection
- === TCP connection

- TCP connection to QUIC stream mapping
The stream mode

<table>
<thead>
<tr>
<th>Client</th>
<th>Concentrator</th>
<th>Final Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>STREAM[0, &quot;TCP Connect, End&quot;]</td>
<td>SYN</td>
<td></td>
</tr>
</tbody>
</table>

Legend:

--- QUIC connection
=== TCP connection

- TCP connection to QUIC stream mapping
- Client initiates a connection with a TLV
The stream mode

Client | Concentrator | Final Destination
-------|--------------|-------------------
STREAM[0, "TCP Connect, End"] | SYN | SYN+ACK
-----------------------------|----|-------------------
STREAM[0,"TCP Connect OK, End"] | <------------------------------| Legend:
|                               |    | --- QUIC connection
|                               |    | === TCP connection

- TCP connection to QUIC stream mapping
- Client initiates a connection with a TLV
The stream mode

<table>
<thead>
<tr>
<th>Client</th>
<th>Concentrator</th>
<th>Final Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>STREAM[0, &quot;TCP Connect, End&quot;]</td>
<td></td>
<td>SYN</td>
</tr>
<tr>
<td></td>
<td>--------------- &gt;</td>
<td>================ &gt;</td>
</tr>
<tr>
<td>STREAM[0,&quot;TCP Connect OK, End&quot;]</td>
<td></td>
<td>SYN+ACK</td>
</tr>
<tr>
<td>&lt;-----------------------------</td>
<td></td>
<td>================ &gt;</td>
</tr>
<tr>
<td>STREAM[0, &quot;bytestream data&quot;]</td>
<td></td>
<td>bytestream data, ACK</td>
</tr>
<tr>
<td>--------------- &gt;</td>
<td></td>
<td>================ &gt;</td>
</tr>
<tr>
<td>STREAM[0, &quot;bytestream data&quot;]</td>
<td></td>
<td>bytestream data, ACK</td>
</tr>
<tr>
<td>&lt;-----------------------------</td>
<td></td>
<td>================ &gt;</td>
</tr>
</tbody>
</table>

Legend:
--- QUIC connection
=== TCP connection

- TCP connection to QUIC stream mapping
- Client initiates a connection with a TLV
- The TCP bytestream is then copied to the QUIC stream
The stream mode

- TCP connection to QUIC stream mapping
- Client initiates a connection with a TLV
- The TCP bytestream is then copied to the QUIC stream
Conclusion

- QUIC can be used to convey many network protocols efficiently.
- There is an interest in considering multihomed clients from the start.
- We defined an application protocol to convey Internet protocols inside QUIC.
  - draft-piraux-intarea-quic-tunnel
  - draft-piraux-intarea-quic-tunnel-session
  - draft-piraux-intarea-quic-tunnel-tcp
- A partial prototype exists as part of [PQUIC], see pquic.org.
- Contributions and collaborations are welcomed at github.com/mpiraux/draft-piraux-quic-tunnel.

[PQUIC] Pluginizing QUIC, Q. De Coninck, F. Michel, M. Piraux et al., SIGCOMM’19
MASQUE

- Protocol defined atop HTTP/3.
- “Impacts on address migration, NAT rebinding, and future multipath mechanisms of QUIC are not anticipated”.

QUIC Tunnel

- Simple binary protocol atop QUIC.
- Considers multihomed devices from the start.
The stream mode

- We propose a one-to-one mapping between a TCP connection and a QUIC stream.
- The Client initiates QUIC streams with a special TLV indicating the final destination.
- Then the TCP bytestream is copied to the QUIC stream data.
- A TLV for indicating a connection failure also exist.